

RECEIVED  
CENTRAL FAX CENTER  
JUN 05 2007

Attorney Docket No.: 40101/00702  
Ref. No.: 2000.020 CON

In the Claims

Please amend the claims as follows:

1-20. (Cancelled)

21. (Previously presented) A method, comprising the steps of:

defining a first memory chunk of system memory, the first memory chunk being a continuous memory space of a first predetermined size;

dividing the first memory chunk into a second memory chunk and a third memory chunk;

assigning control of the second and third memory chunks to a chunk pool;

transferring control of the second memory chunk from the chunk pool to a unit pool;

separating the second memory chunk into units of a second predetermined size;

and

storing data of the second predetermined size in the units.

22. (Previously presented) The method of claim 21, further comprising the steps of:

transferring control of the third memory chunk to a second chunk pool; and

dividing the third memory chunk into a fourth memory chunk and a fifth memory chunk.

23. (Previously presented) The method of claim 21, further comprising the step of:

re-combining the second memory chunk and the third memory chunk to form the first memory chunk.

24. (Previously presented) The method of claim 21, wherein the second and third memory chunks are each one-half the first predetermined size.

Attorney Docket No.: 40101/00702  
Ref. No.: 2000.020 CON

25. (Previously presented) The method of claim 24, wherein the first predetermined size is one of 1,024 bytes, 2,048 bytes, 4,096 bytes, 8,192 bytes, 16,384 bytes and 32,768 bytes.

26. (Previously presented) The method of claim 21, further comprising the steps of:  
transferring control of the third memory chunk from the chunk pool to a second unit pool;  
separating the third memory chunk into units of a third predetermined size, the third predetermined size being different from the second predetermined size; and  
storing data of the third predetermined size in the units of the third predetermined size.

27. (Previously presented) A system, comprising:  
a first chunk of system memory, the first chunk being a continuous memory space of a first predetermined size;  
a plurality of chunk pools, each chunk pool assigned to manage chunks of a predetermined size; and  
a plurality of unit pools, each unit pool assigned to manage memory units of a unit size, wherein the first chunk is divided into a plurality of chunks, each of the plurality of chunks being assigned to the corresponding chunk pool and wherein one of the chunk pools transfers control of one of the chunks to one of the unit pools which divides the chunk into the memory units of the unit size, the corresponding unit pool storing data of the unit size in the units.

28. (Previously presented) The system of claim 27, wherein the system memory is one of static random access memory, dynamic random access memory, synchronous dynamic random access memory and enhanced dynamic random access memory.

Attorney Docket No.: 40101/00702  
Ref. No.: 2000.020 CON

29. (Previously presented) The system of claim 27, wherein the one of the unit pools recombines the memory units into the chunk and passes control of the chunk back to the corresponding chunk pool.
30. (Previously presented) The system of claim 27, wherein, when the memory units are full, the one of the unit pools requests the one of the chunk pools to transfer control of a second chunk to the one of the unit pools to divide the second chunk into additional memory units.
31. (Previously presented) The system of claim 27, wherein the first chunk of system memory is used exclusively for data to be stored by the unit pools.
32. (Previously presented) The system of claim 27, wherein the first chunk of system memory is assigned during system initialization.
33. (Previously presented) The system of claim 27, wherein the chunk pools include a hierarchical relationship where chunk pools receive chunks from a next highest chunk pool in the hierarchy.
34. (Previously presented) The system of claim 33, wherein chunk pools return unused chunks to the next highest chunk pool in the hierarchy when a sum of the returned chunks equals the predetermined size of chunk controlled by the next highest chunk pool.
- 35-40. (Cancelled)